CSC420 Project report
Detection and classification of cells in medical images

Luyuan Chen, 999516082
Ming Yue, 999962018 (Partner)

1. Introduction

The project of selection is using computer vision techniques to detect and classify the cells and nuclei in properly stained (with *hematoxylin* and *eosin*) and captured medical images. This project involves using both traditional computer vision methods and machine learning approaches. We chose the project because of its infusion of modern image processing techniques (by using convolutional neural network) and math-toward vision practise.

The machine learning approach used employed convolutional neural network based on [1], which is derived from the *LeNet* implementation from LeCun et.al.[2]. There are many proposed computer vision approaches, including blending CMY image channels and using hough transform [3](E. Cossato, 2008), using modified circular hough transform (CHT) [4] and using a generalized hough transform[5]. To augment the data, the aforementioned approaches all implement image pre-processing such as colour channel blending, using Gaussian filters, sliding window on image pyramids, blob detection and most importantly, hough transform.

I am solely responsible for the detection part and my partner (Ming Yue) is independently responsible for the classification model. This report focuses on the detection part.

2. Methods

Computer vision approach

After inspecting and researching about the data set provided, I made several assumptions about the images:

1. The cell nuclei and cytoplasm reacts differently to the stain and will exhibit different colour in the images
2. The cells are mostly in circular or elliptical shapes and similar in size

The first step is then to separate out the nuclei and cytoplasm according to their colour. The colour difference comes from the different reaction to the stain: hematoxylin binds to nuclear chromatin and eosin binds to cytoplasmic components [3]. Furthermore the stains react differently to light so that the optical density captured in the images will differ. So that by project the image onto the vector that represents the reaction of specific stain (colour deconvolution), the cell can be discovered[6]. For the greyscale image approach, I re-weighted the image so that the

red and blue channels (which made up the purple colour) are more prominent in the adjusted image.

Here is a demonstration of colour channel transformation

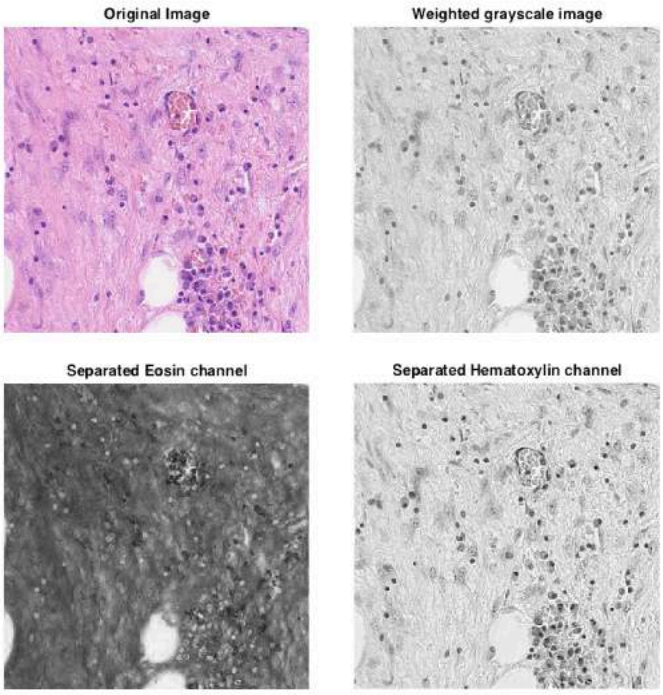

Figure 1: Different image color blending method

Next, I calculated the E channel's gradient using Sobel method and then detected the edge by Canny edge detector with threshold 0.3. The cells will have clearer shapes with edges detected.

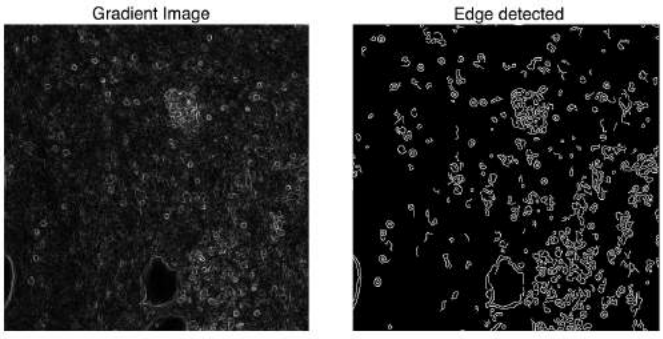

Figure 2: Gradient image and detected edges

After getting the image edge, I used a simple form of voting inspired by generalized hough transform and voting. That is, when one point on the edge is encountered, I add 1 to all the

voting within the certain range to that point (on the voting image). The range used is in circular shape with radius 5. This way, the centre of the irregular shape will receive most of the voting as the disk will likely to pass over the centre multiple times. After this step, the cell centres should be obvious. Plotted below are the examples of 1) Voting on one cell and 2) voting on all cells.

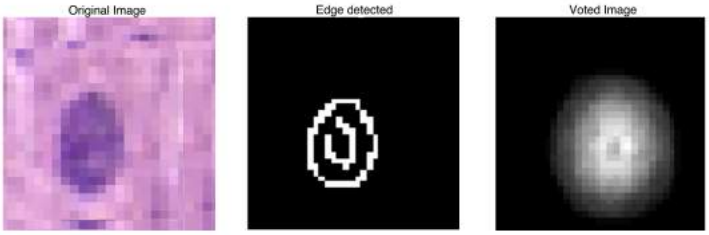

Figure 3: Original image, detected edge and voting of one cell

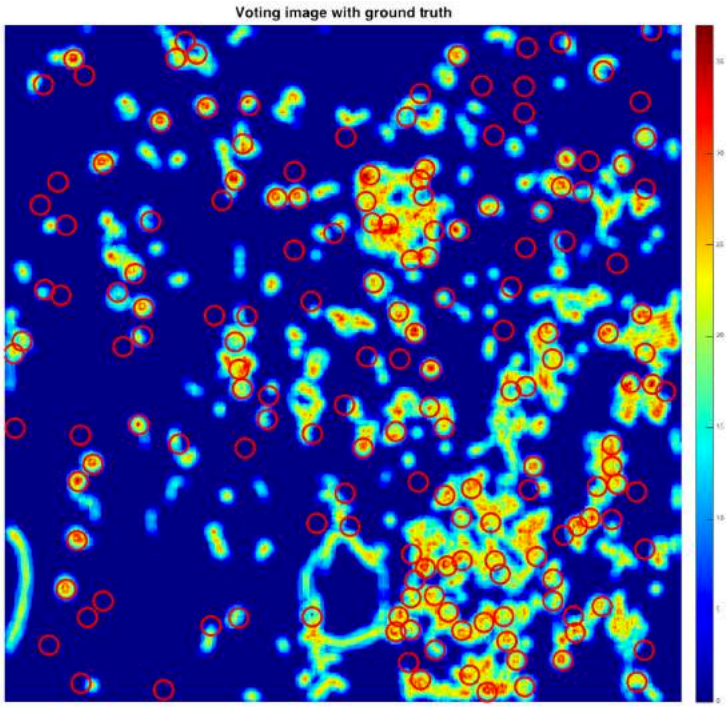

Figure 4: Detected peaks in voting with ground truth cells

The final step is to use non-maximal suppression to separate out the local maxima, which will be centres of the cells.

Machine Learning approach

The code for machine learning approach is based on Cifar10 code provided by Google Tensorflow [7]. The model will classify whether a cell is present at the centre of a 27×27 patch. The

input to the model is the image patch while the output is a one-hot encoded classification.

The convolutional model used is as follows:

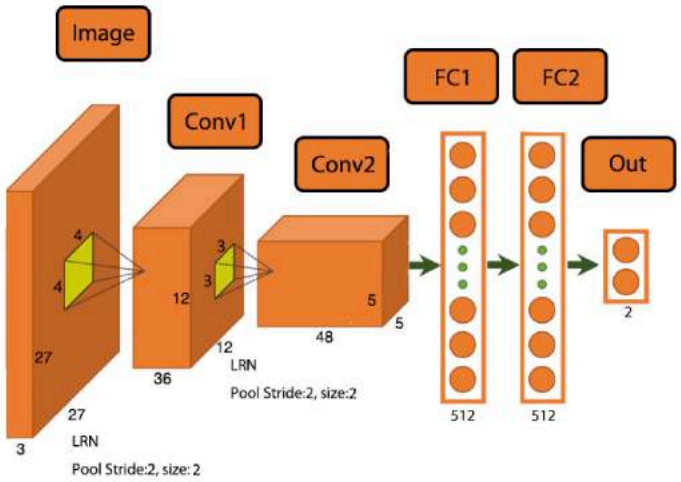

Figure 5: Structure of CNN used

The size chosen for each layer is the same in the paper [1] as listed:

Layer	Size
Convolution 1	$4 \times 4 \times 1 \times 36$
Convolution 2	$3 \times 3 \times 36 \times 48$
Fully connected 1	1200×512
Fully connected 2	512×512
Output	512×2

Table 1: Parameter of the neural nets

Pooling layers with size 2×2 and stride 2 is added after each convolutional layer. Also, to prevent overfitting, the *local response normalization* is added after convolutional layers as well, this is inspired by AlexNet [8]. On the output layer, Softmax is used to provide a probabilistic interpretation of the feature discovered by the convolutional layers.

For the training images, the image with a cell in centre is carved out of the original image in 27×27 patches. Moreover, to train this binary classification model property, one image of false example is generated for each positive image. There are about 50 thousand image patches generated and I used 69% of them for training and the remaining 13% for cross validation and 18% for the final testing. Here's an example of images with negative and positive example:

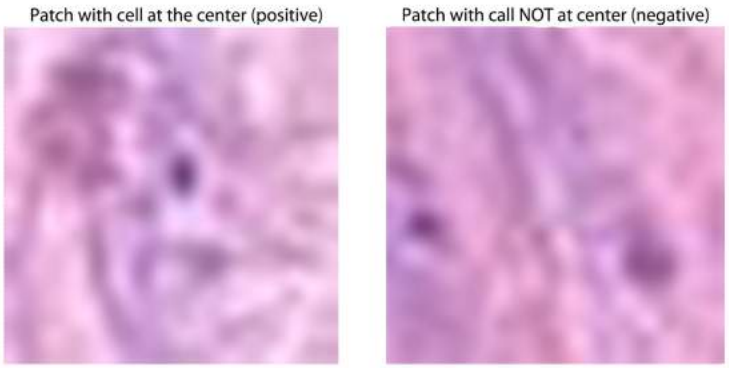

Figure 6: An example of images with positive(left) and negative(right) examples

To fight overfitting, the images are randomly pre-processed prior to feeding. The precessing includes random cropping, flipping and distortion the image patch. The images are shuffled in a batch of 128 images during training. Further, dropout with a keep rate 0.5 is added in the fully connected layers to enhance generalization during training. Adam optimizer is used in training for a faster converging rate.

3. Main challenges

I think the most challenging parts of computer vision method are when finding a better colour blending method and testing possible variant of detection techniques. Initially I tried blending RGB, CMY and La*b* colour spaces in one image.The hard part is the ratio is tricky to get right. One combination might work well for nuclei but not for cytoplasm. In [3], I found that I could use two images: one for each stain. I then found a way to separate the stain by the optical density as described in the colour deconvolution paper [5].

On the detection part, the challenge is to find a proper way to detect nuclei. First I tried simple blob detection as well as SIFT method on the colour blended images, but they did not provide promising result. I then tried using HoG detectors and it can't provide confident enough descriptors given the size of the patch and image. Later I discovered using gradient/edge detection can enhance the performance. I found out there are a lot papers using different hough transform to detect cells. I tried using circular and elliptical hough transforms, but many cancerous cells are not in analytical shapes. Last, I tried a rather simple hough voting method that follows the edge to detect the centre of irregularly shaped objects. The result is satisfactory, as shown in figure 3.

The main challenge in machine learning approach is coping with low validation accuracy. I firstly tried a simple CNN design with greyscale image. The model converged rapidly but the precision on the validation set is unsatisfactory (roughly 49%, which is bad for a binary classification problem). One solution is found by reading papers and online articles about how to minimize overfitting in convolutional neural networks. I chose the dropout method because of its simplicity[9]. The accuracy increased to about 59% but it's not enough. Eventually I realized

the simplicity of the model prevented it to perform better (with grey scale images). The final solution which uses colour images provided much better result.

4. Result and discussion

Computer vision approach

I tried several methods in each step of the processing stage. As previously discussed, the simple colour blending technique (like `rgb2gray` in Matlab) did not perform well through later stages. By using the colour deconvolution and RGB re-weighting method, the cells can be much better separated out. Image gradient calculation is added before the edge detection stage. It improved performance by reducing the false negative results. The reason is, running edge detection on raw images will likely to discover many inter-cellular structures and generate many unwanted edges. It will also miss many cells resulting in false negative. By taking a gradient, the edges of the cells will be much clearer in that the nuclei structure can be detected (clear in the figure below). The more detailed edge detection will aid the hough transform in the next step.

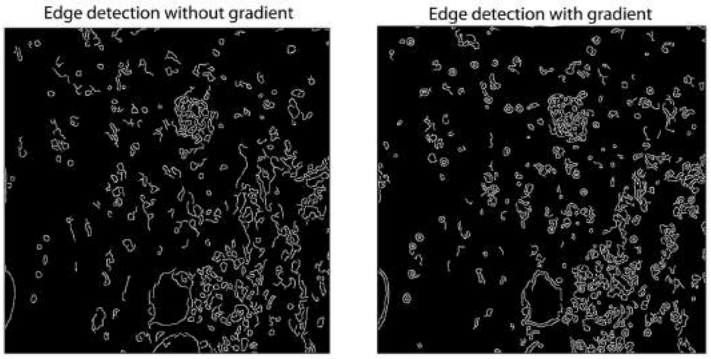

Figure 7: Comparison of edge detection with and without calculating gradient

The voting steps is done by adding vote to accumulator image when an edge is encountered. The mask for the accumulation is 5 pixels.

Also, for images with many cells in proximity with others, the size of voting radius needs to be changed so that the voting won't overlap. If the voting overlaps, the local maxima is likely to appear at the middle of two cells rather than at their own centres. The solution is to decrease the voting radius so that the disks don't overlap. Figure 8 is an example of using different voting radius. As shown, in areas many cells are present, the high peaks in voting prevented the discovery for single cells.

The final step is non-maximal suppression. Because the fact that the voting image will have several peaks in one cell, and the limited NMS radius constrained by the cells size, I averaged the image prior to applying NMS. The averaging method of choice is a 4×4 gaussian filter with $\sigma = 0.3$. Shown below is the result after NMS. The circles represent the ground truth

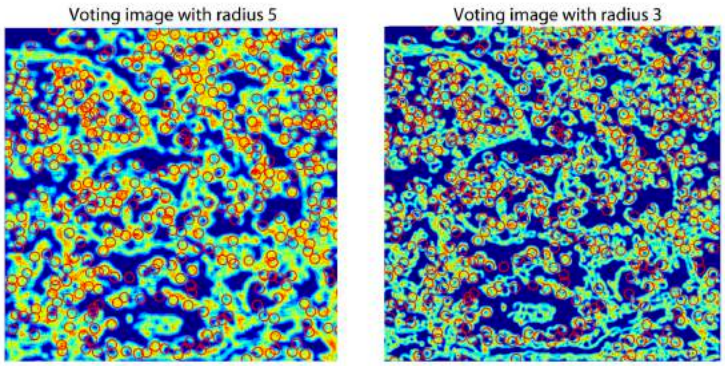

Figure 8: Comparison of different voting radius: left(5) right (3)

cell locations and the white dots are the discovered cell locations. The combination of voting, gaussian averaging and NMS radius has to be further tuned to eliminate the false positive and false negative locations. At current stage, if there are white dots in the red circles, I characterize it as discovered.

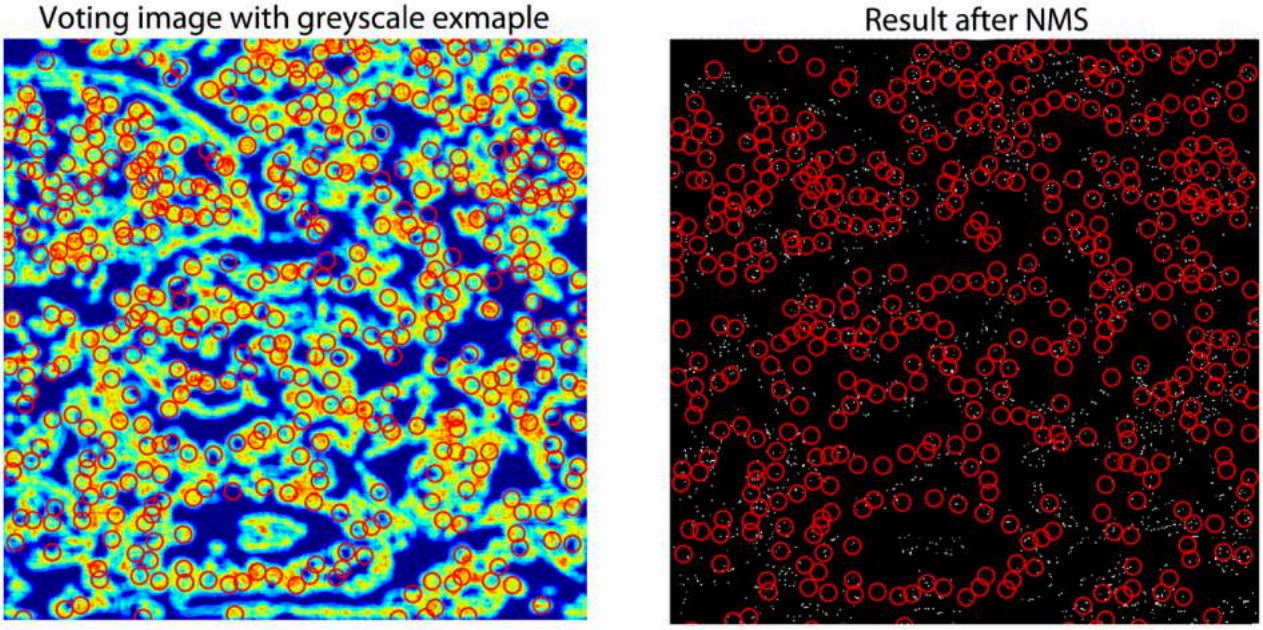

Figure 9: Voting using greyscale image and the final result after NMS

The detection method is robust when having a closed (or mostly closed) contour, but not with a rod shape cells. That is because, the model voting accumulates mostly on the long edge so that it would miss the centre, and if the response along the edge is strong, it's likely to generate many false positive entries. In the images with clearer and not very densely packed cells, the model is able to find 61% of the cells, as shown in image 4.

Further, the model will miss the cells which are faint in colour in the stained images. This is expected though, because that if the cell is of interest, the researcher would use a better staining method to highlight it.

Machine learning approach

I used TensorFlow to train the machine learning model. Firstly I used greyscale images to train the neural nets and the training accuracy is very high (close to 100%). However, when I run the model with testing data, the precision is only about 50%. After adding the dropout and other components, the accuracy of training increased to about 60%. Eventually I used colour images for training and the final testing precision is about 89%. The result from TensorBoard is shown below:

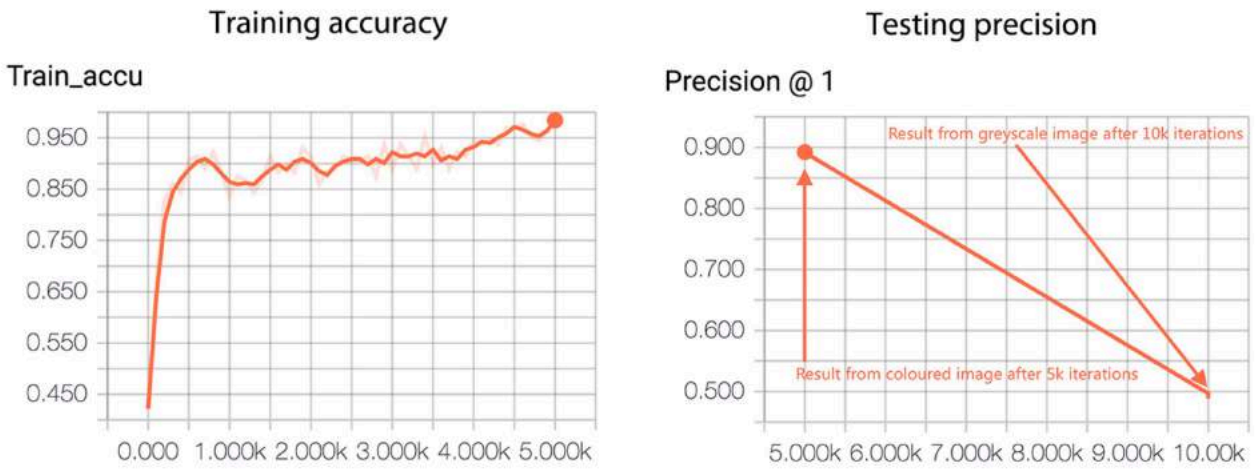

Figure 10: Training accuracy and testing precision

Based on the output from the test set, the performance are summarized in the table below

Item	Value
Precision	89.5%
Recall	70.3%
F1 Score	78.7%

Table 2: Model performance

As compared to the paper, the precision our model achieved is much higher (89.5% vs. 69.7%). On the recall side, the result is quite similar (70.3% vs. 68.7%). Overall, the F1 score of our

model (78.7%) is about 10% higher than the CP-CNN presented in the paper, whose F1 score is 69.2%. Our model's performance is comparable to the SC-CNN method presented in the paper. Intuitively, 89 out of 100 patches are predicted correctly among all predicted positive patches and only 80% of the actual positive images are predicted to have cell in the centre.

The high precision and low recall combination shows that the model is confident when the prediction is positive. This is because the false patches are generated randomly, so that many of the generated patches do not have a cell in the patch. The filter banks will be trained very well to detect features at the centre. But the model is less robust if the cell's position shifts, or the cell is of irregular shape (such as the lengthy shapes discussed in computer vision section). This will explain the comparatively low recall value. Another factor that contributed to the high precision is the adding of regularization methods. Dropout and other approaches are added to prevent overfitting such that when the cells are in regular shape, the model will always be confident.

5.Conclusion and future work

In the detection part of the project, both computer vision and machine methods are used. The computer vision approach employed techniques like colour deconvolution, gradient image, edge detection and non-maximal suppression(NMS) to produce meaningful result. Coloured image and greyscale image from manual re-weighting did equally well in the detection logic. The model is robust with regular shaped and properly stained cells. Future improvement of the computer vision part involves tuning the parameter of gaussian smoothing, voting radius and NMS radius. One could also try implementing a more complexed voting function.

In the machine learning part, I used a 5 layer convolutional neural network to classify whether a cell in present in the middle of a patch. Dropout, local response normalization and distorted image feed is added to prevent overfitting. In coloured images, the model has very high precision and lower recall: it predicts with confidence but might miss some positive patches. Overall it has a F1 score of 78.7%. However, the model is not robust with greyscale images, it only achieved lower than 60% accuracy in test set. Future improvement can be generate a better training set: false examples should be more complexed. This will improve the recall performance of the model.

References

[1] K. Sirinukunwattana, S. E. A. Raza, Y.-W. Tsang, D. R. Snead, I. A. Cree, and N. M. Rajpoot, "Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1196–1206, 2016.

[2] Y. Le Cun, L. Jackel, B. Boser, J. Denker, H. Graf, I. Guyon, D. Henderson, R. Howard, and W. Hubbard, "Handwritten digit recognition: Applications of neural network chips and automatic learning," *IEEE Communications Magazine*, vol. 27, no. 11, pp. 41–46, 1989.

[3] E. Cosatto, M. Miller, H. P. Graf, and J. S. Meyer, "Grading nuclear pleomorphism on histological micrographs," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pp. 1–4, IEEE, 2008.

[4] M. Smereka and I. Duleba, "Circular object detection using a modified hough transform," *International Journal of Applied Mathematics and Computer Science*, vol. 18, no. 1, pp. 85–91, 2008.

[5] A. Thomas, T. Davies, and A. Luxmoore, "The hough transform for locating cell nuclei," in *Applications of Image Processing in Mass Health Screening, IEE Colloquium on*, pp. 8–1, IET, 1992.

[6] A. C. Ruifrok, D. A. Johnston, *et al.*, "Quantification of histochemical staining by color deconvolution," *Analytical and quantitative cytology and histology*, vol. 23, no. 4, pp. 291–299, 2001.

[7] G. T. Team, "Convolutional neural networks," *TensorFlow tutorials*.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[9] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.